

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA



CENTRALIZAÇÃO DE CONFIGURAÇÕES DE SERVIÇOS NUM AMBIENTE DE MICRO-SERVIÇOS

Bernardo Filipe Martinho Neves

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Engenharia de Software

Versão Pública

Dissertação orientada por:
Prof. Doutor Alcides Miguel Cachulo Aguiar Fonseca

2020

Agradecimentos

Gostaria de começar por agradecer ao meu supervisor, Professor Alcides Fonseca, pela oportunidade de trabalho neste projecto para a minha dissertação e para a supervisão do mesmo.

Um enorme agradecimento especial ao João Rebelo por toda a paciência e apoio que me foi prestado ao longo do projecto, bem como aos restantes membros da equipa da Profidata.

Aos meus pais e à minha família, uma palavra de gratidão. por todo o apoio e incentivo que me deram desde o primeiro dia desta viagem.

Por fim um obrigado a todos os meus amigos e colegas da faculdade por partilharem comigo esta viagem durante estes ultimos longos anos.

Resumo

Configurações são um conceito bastante simples, afinal de contas podem ser apenas pares de chave-valor. Mas se pensarmos bem são elas que moldam quase tudo o que nos rodeia, pelo menos no que é manufaturado, como uma caneta, por exemplo, que pode possuir diferentes cores, pois a cor da qual uma caneta escreve é uma das suas configurações, sendo um simples par de chave-valor, onde uma caneta azul possui a chave cor correspondente ao valor azul nas suas configurações. Contudo nem sempre é assim tão simples, aliás em sistemas informáticos as configurações podem ficar bastante complexas e confusas, sendo que quanto maior o sistema maior o número de configurações que este necessita.

Nos últimos anos bastantes empresas que desenvolvem sistemas informáticos, por diversos motivos, têm decidido mudar a arquitectura dos seus sistemas para Micro-Serviços. No entanto, com esta mudança também surgiram novos problemas. Será que devemos colocar todas as configurações de um serviço centralizado ou distribuído pelos vários serviços? Será que devemos manter os ficheiros de configurações locais a cada serviço ou externas? Estas foram algumas das perguntas mais frequentes nestas transições. A resposta na maioria dos casos é a utilização uniforme de configurações externas de forma a não fragmentar as configurações pelos vários serviços.

Utilizar configurações externas permite assim uma melhor gestão das mesmas, no entanto ainda existem pontos que podem ser melhorados. Um desses pontos é o número de configurações necessárias e como estas interagem umas com as outras. Ao utilizar uma arquitectura de Micro-Serviços estas configurações são ainda em maior quantidade, do que num sistema monolítico por exemplo, pois cada serviço necessita não só das suas próprias configurações, mas também de configurações mais gerais do sistema. Isto pode levar a configurações duplicadas, sendo assim difíceis de corrigir ou alterar caso necessário visto que uma configuração pode ter de ser modificada em diferentes locais.

A solução aqui apresentada tenta minimizar estes problemas, adicionando um serviço de gestão de configurações independente do produto. Esta proposta irá possuir um sistema hierárquico de configurações, para assim diminuir o número de configurações duplicadas. Também estará provida de um sistema de versões, possibilitando assim uma fácil detecção de mudanças, um mecanismo para voltar a um estado anterior e uma forma de rever alterações antes destas serem aplicadas.

Palavras-chave: Configurações, Micro-Serviços, Hierarquia de Configurações, Resolução de Variáveis

Abstract

Configurations are a simple concept, for they are just key-value pairs. Despite being a simple concept, they shape almost everything around us, at least what is manufactured, such as a pen, for example, that may have different colors: the color of which a pen writes is one of its configurations. a simple key-value pair, where a blue pen has the key color corresponding to the value blue in its configurations. However it is not always so simple, in computer systems configurations can be very complex and confusing, and the larger the system, the greater the number of configurations it needs.

In recent years many companies that develop computer systems have, for various reasons, decided to change the architecture of their systems to Micro-Services. However, with this change new problems arose. Should we put all the configurations in one place or in multiple places? Should we keep the local configuration files for each service or external? These were some of the most frequently asked questions in these transitions. The answer in most cases is the uniform use of external configurations so as not to fragment the configurations across the multiple services.

Using external configurations allows for a better management, however there are still points that can be improved. One of these points is the number of configurations required and how they interact with one another. When using a Micro-Services architecture, these configurations are still in greater quantity than in a monolithic system, for example, because each service requires not only its own configurations, but also more general configurations of the whole system. This can lead to duplicate configurations and therefore more difficult to fix or change, since they can be in different files in different folders.

The solution presented here attempts to minimize these problems by adding a configuration service independent from the product. This proposal features a hierarchical system of configurations in order to reduce the number of duplicate configurations. It will also be provided with a versioning system, allowing for easy detection, reviewing and reversion of changes.

Keywords: Configurations, Micro-Services, Hierarchy of Configurations, Variable Resolution

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Objectivos	2
1.3	Estrutura do documento	3
	Bibliografia	5
	Índice	6

Capítulo 1

Introdução

1.1 Motivação

O Xentis[5], o sistema no qual foi realizado o projecto, é um produto que está a migrar de uma estrutura monolítica para Micro-Serviços devido às vantagens oferecidas por esta arquitetura, nomeadamente no que toca a publicação de serviços, escalabilidade, bem como resiliência do produto. A facilidade de publicação de serviços permite uma entrega contínua dado que cada um pode ser publicado individualmente. Isto permite que diferentes ferramentas e/ou tecnologias sejam usadas em cada serviço, assim como o desenvolvimento de serviços em paralelo por equipas diferentes, agilizando assim este desenvolvimento individual dos serviços. Uma das grandes vantagens dos Micro-Serviços é a sua escalabilidade independente, sendo que cada serviço pode ser escalado facilmente sem alterar os restantes. Esta independência entre serviços leva a uma grande capacidade de resiliência do produto, sendo que se ocorrer uma falha num serviço, esta não afeta os restantes, a não ser que este seja um serviço crítico.

Mas com a utilização de Micro-Serviços também surgem novos desafios, sendo um deles a gestão de configurações. Existem duas formas principais de resolver este desafio.

Uma forma, sendo esta a tradicional, em que cada serviço possui os seus próprios ficheiros de configurações e que podem ou não conter configurações extra sobre outros serviços do qual este depende ou interage. Desta forma serviços diferentes que dependem dos mesmos serviços externos ou internos ao produto irão conter configurações duplicadas, como por exemplo, configurações de comunicação entre serviços ou de acesso a bases de dados. Neste caso a alteração de configurações pode ser bastante difícil considerando que tem de ser realizada em múltiplos ficheiros em múltiplas localizações.

Outra forma, em que um serviço central é responsável pela gestão de configurações, para que outros consigam aceder às configurações que necessitam. Isto permite que a mecânica interna deste serviço central seja independente dos restantes serviços, bem como a diminuição de configurações duplicadas e a facilidade de alteração e gestão dos ficheiros. Se se considerar a existência de uma interface gráfica para a gestão de configurações,

o serviço central irá simplificar a forma como a interface obtém essas configurações, uma vez que não irá necessitar de conhecer todos os serviços do produto que façam gestão de configurações. Adicionalmente o serviço central também poderá retirar alguma complexidade que os outros serviços teriam no caso de serem estes a fazer esta gestão. Contudo este serviço irá tornar-se num serviço crítico do produto do qual os restantes serviços irão depender.

1.2 Objectivos

O Xentis possui uma solução de configuração ad-hoc onde cada serviço tem os seus próprios ficheiros de configuração, bem como outras soluções que podem ser mudadas para uma solução centralizada e assim ter uma solução mais vantajosa em termos de custo de desenvolvimento e operação no futuro. Com isto este projeto consiste nesta solução centralizada de gestão de configurações, onde o principal objetivo é o desenvolvimento de um serviço para este propósito.

Como o Xentis é um produto que auxilia a gestão de investimentos tem de seguir várias questões legais, isto faz com que seja necessário a existência de um histórico de todas as suas alterações para se poder recriar situações específicas, que podem ser requeridas em avaliações, auditorias e estudos destes investimentos. Para além disso, este histórico pode ser ainda usado para replicar problemas que tenham existido em clientes ou revisões e assim facilmente colocar o produto nas mesmas situações em que o problema ocorreu, tornando o processo de resolução de erros mais rápido. Outro mecanismo utilizado no desenvolvimento do Xentis é a revisão de alterações antes de serem publicadas, isto inclui configurações de modo a evitar erros e prevenir possíveis falhas no futuro.

Com isto, este serviço terá como objetivos fornecer as configurações necessárias a cada outro serviço do Xentis, bem como disponibilizar ações de gestão das mesmas, tal como adicionar, modificar ou remover configurações, através de uma interface gráfica tanto no Xentis CLI como numa Web GUI.

O serviço será ainda provido de um módulo de resolução de variáveis, algo já existente nas configurações antigas. Estas podem referenciar variáveis de ambiente, propriedades do próprio ficheiro ou de outros ficheiros. Esta funcionalidade tem de conseguir resolver sequências de variáveis, onde uma variável pode conter outra, desde que não provoque ciclos. Para além disso, tem de saber resolver operações lógicas, como condicionais. Mais operações podem vir a ser adicionadas no futuro para facilitar as configurações, com isto, a solução tem de ser extensível para facilmente adicionar novas formas de resolução de variáveis. Para finalizar, terá ainda um módulo de versionamento de configurações, que suporte comparações entre versões, importação e exportação de versões, bem como possibilitar a criação de um histórico baseado nas versões. Este serviço terá que suportar também diferentes ambientes de configurações, dando a possibilidade de existir ambientes

de desenvolvimento, teste e de publicação, isto facilita assim a mudança de ambiente nas várias fases do produto. O armazenamento dos ficheiros de configurações será estruturado de forma hierárquica, de modo a possibilitar diferentes ambientes de configurações e diferentes níveis de especificidade das configurações.

Este projeto foi realizado numa equipa de dois elementos no qual foram trabalhadas as componentes de gestão das configurações, bem como a persistência e estrutura destas, sendo que a componente gráfica foi realizada pelo outro elemento da equipa.

1.3 Estrutura do documento

Este documento está dividido em sete capítulos, sendo o primeiro uma introdução ao projecto descrevendo os motivos e objectivos do mesmo, o segundo capítulo com o resultado do estudo de trabalhos relacionados, no terceiro uma descrição do plano do trabalho, no quarto é apresentado o trabalho realizado bem como os desafios encontrados, no quinto capítulo são discutidas as conclusões retiradas do projecto, o sexto capítulo com a bibliografia e um sétimo capítulo com anexos ao documento.

Bibliografia

- [1] Algoritmo de balanço de parênteses numa expressão. <https://www.geeksforgeeks.org/check-for-balanced-parentheses-in-an-expression/>.
- [2] Biblioteca java spring cloud config. <https://spring.io/projects/spring-cloud-config>.
- [3] Biblioteca java vert'x config. <https://vertx.io/docs/vertx-config/java/>.
- [4] Manual pro git. <https://git-scm.com/book/en/v2>.
- [5] Produto da profidata group. <https://www.profidatagroup.com/xentis/>.
- [6] Spring boot 2.0 properties binding. <https://spring.io/blog/2018/03/28/property-binding-in-spring-boot-2-0>.
- [7] Akash Bhingole. Microservices - central config server layer. *DZone*, Feb. 2018.
- [8] Jknack. Biblioteca java handlebars, logic-less and semantic templates with java. <https://jknack.github.io/handlebars.java/>.
- [9] Norbert Potocki. Biblioteca java cfg4j. <https://www.cfg4j.org>.
- [10] Matthias Sohn. Biblioteca java de interação com git. <https://www.eclipse.org/jgit/>.

